

AD-A047 907

NORTHROP CORP HAWTHORNE CALIF ELECTRONICS DIV
AN/BRN-7 COMPUTER PROGRAM SPECIFICATION. VOLUME II. DESIGN SPEC--ETC(U)
OCT 73

N00039-73-C-0209

F/G 17/7

NL

UNCLASSIFIED

NORT-73-48-VOL-2

AD
A047907



END
DATE
FILMED
78
DDC



①

NORTHROP

Electronics Division

AD-A047907

- ☐ This submittal applies to AN/BRN-7 (Submarine Ω) only.
- ☐ This submittal applies to AN/SRN-() (Hydrofoil Ω) only.
- ☒ This submittal applies to both AN/BRN-7 and AN/SRN-().

CONTRACT NO: N00039-73-C-0209

PROGRAM NAME: AN/BRN-7

CDRL No: A01D, A01E, A01F

Title of CDRL: Computer Program Design Specification
Computer Subprogram Design Document
Data Base Design Document

Title of DOC: AN/BRN-7 Computer Program Specification
NORT 73-48
Volume 2 thru 13

Date: 1/16/74

Initial Submittal: <input checked="" type="checkbox"/>	Release	AP S 1-16-74
Resubmittal: <input type="checkbox"/>	Authentication	

Prepared by: System Integration Unit
Orgn. Name

A4734 X574
Orgn. No. and Ext.

Approved by:

DDC
RECEIVED
DEC 23 1977
D

Norman O. Clark
Supervisor of Preparing
Orgn.

Bill Zylar
Program Mgr.

W. J. Oman
Proj. Engr./Proj. Mgr.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

NORTHROP

Electronics Division

Copy No. 2

NORT 73-48

**AN/BRN-7 COMPUTER
PROGRAM SPECIFICATION**

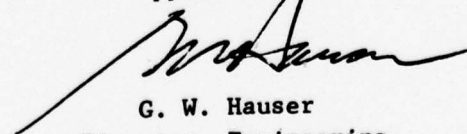
Volume II

DESIGN SPECIFICATION
October 12, 1973

ADDITION FOR	
RTG	Write Section <input checked="" type="checkbox"/>
DDG	Butt Section <input type="checkbox"/>
WHANNOUNCED	<input type="checkbox"/>
IDENTIFICATION	
Per DDC Form 50 on file	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist. Avail. and/or Special	
A	

Page 1 is missing
For Scope, see Vol. I

Approved by



G. W. Hauser
Director, Engineering
Navigation Department

**Volume II
of the
AN/BRN-7 OMEGA COMPUTER
PROGRAM SPECIFICATION**

Volume

- I Performance Specification**
- II Design Specification**
- III Synchronization Subprogram Design**
- IV OMEGA Processing Subprogram Design**
- V Tracking Filter Subprogram Design**
- VI Kalman Filter Subprogram Design**
- VII Propagation Prediction Subprogram Design**
- VIII Navigation Subprogram Design**
- IX Executive Subprogram Design**
- X Control-Indicator Subprogram Design**
- XI Built-in Test Subprogram Design**
- XII Common Subroutines Subprogram Design**
- XIII Appendix**

CONTENTS

<u>Section</u>	<u>Page</u>
1 SCOPE	1
2 APPLICABLE DOCUMENTS	2
3 REQUIREMENTS	3
3.1 Review of Functional Requirements	3
3.2 Dependence of Requirements on Timing	6
3.3 Subprogram Description and Allocation of Requirements	8
3.4 Executive Routine	13
3.4.1 Interrupts	13
3.4.1.1 External Interrupts	13
3.4.1.2 Internal Interrupts	13
3.4.2 Interleaved Program Sequencing	14
3.4.3 Main Program vs Five Millisecond Interrupt Routine	15
3.4.3.1 Start-Up	15
3.4.3.2 OMEGA Navigation Iteration	20

SECTION 2**APPLICABLE DOCUMENTS**

- a) NORT 71-71, Vol I, Computer Program Performance Specification
- b) OPERATOR'S AND MAINTENANCE MANUAL, Submarine OMEGA Navigation Set, AN/BRN-7
- c) NORT 71-71, NDC-1070 MACRO Assembler, May 1971
- d) NORT 68-115A, Detailed Description of NDC-1070 Computer Instructions, Revision A, February 1970.
- e) NORT 69-87A, NDC-1070 FLOW CHART Program User's Manual.

SECTION 3

REQUIREMENTS

3.1 REVIEW OF FUNCTIONAL REQUIREMENTS

The functional requirements generated in the Computer Program Performance Specification are summarized in Table 1, and the sequential interdependence of the functions is shown in Figure 1. Each function of the diagram must be initiated at a particular point in the sequence, and many must be initiated at a particular point in time, sometimes both.

TABLE 1 REVIEW OF FUNCTIONAL REQUIREMENTS

<u>Section*</u>	<u>Title</u>	<u>Descriptive Summary</u>
3.3.1	Synchronization	To synchronize the timing of the computer program functions to that of the incoming OMEGA signals.
3.3.2	Signal Input Timing	A multiplex function to direct incoming: <ul style="list-style-type: none"> • Coherent pulses to 3.3.1 Synchronization • Noise pulses to 3.3.5 Noise Calculations • Test pulses to 3.3.4 Bias, Scale Factor, etc. • Burst pulses to 3.3.7 Burst Phase Measurement
3.3.3	Antenna Switching Control	Provides computer program control of antenna loop/float inputs to receiver, optimizing signal-to-noise ratio.
3.3.4	Bias, Scale Factor and Phase Shift	Provides calibration which removes hardware-induced phase shifts.
3.3.5	Noise Calculations	Provides means of assessing signal credibility values based on signal-to-noise ratio (SNR).
3.3.6	Phantom Calculations	Provides means of removing spurious, hardware-induced signals of low signal strength.
3.3.7	Burst Phase Measurement	Calculates first phase measurement from stations and removes all phase shifts noted above. Also calculates variance (credibility) of that measure based on SNR.
3.3.8	Base Station Selection	Selects that station of greatest signal strength against whose phase measurement all others will be differenced.
3.3.9	Phase Difference Processing	Differences phases measurement from each station against that of base station and computes the difference variance.

*Section number refers to Volume I, Performance Specification.

10802-1

TABLE 1 REVIEW OF FUNCTIONAL REQUIREMENTS (continued)

3.3.10 Tracking Filter	Statistically averages each phase difference measurement and recomputes each phase difference variance until variance is within the acceptance criterion of the Combinational Filter.
3.3.11 Combinational Filter	A Kalman filter which synthesizes Tracking Filter data with phase velocity data from Propagation Prediction, and provides corrections to position, velocity and OMEGA oscillator parameters.
3.3.12 Propagation Prediction	Provides Combinational Filter with real-world propagation data.
3.3.13 Velocity and Heading Processing	Smooths velocity and heading inputs from peripherals and corrects velocity with data from Combinational Filter.
3.3.14 Navigation	Provides OMEGA system with present position in form of R_{ij} matrix. Uses Vel. Processing, Combinational Filter, and Control-Indicator to update.
3.3.15 Control-Indicator	Interfaces OMEGA data with operator.
3.3.16 Built-In-Test Procedures	Assesses functional capability of hardware on both operational and diagnostic levels.
3.3.17 Built-In-Test Equipment	Assesses functional capability of hardware which cannot be tested by 3.3.16 BIT.

10802-2



FIGURE 1 FLOW DIAGRAM OF FUNCTIONAL REQUIREMENTS

3.2 DEPENDENCE OF REQUIREMENTS ON TIMING

In Volume I, Computer Program Performance Specification, each burst and slot of the OMEGA pattern was segmented into four periods to eliminate transients and signal anomalies at the rise and fall time of the bursts. This is reproduced in Figure 2 below.

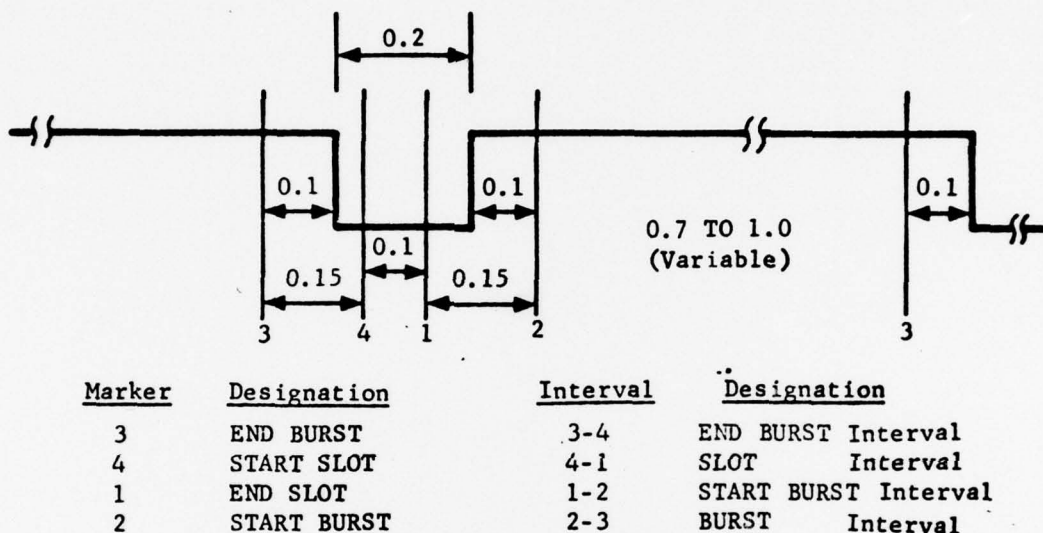


FIGURE 2 BURST AND SLOT SEGMENTATION

There are six functional requirements whose execution is directly dependent upon the burst and slot segmentation. There are others which are indirectly dependent; indirectly in that they are sequentially dependent upon data from a function which is directly dependent. This is illustrated in Figure 3, whose purpose is to give the reader a feeling for the eventual allocation of functional requirements to the computer routines and subprograms. As indicated in the figure the segmentation of each burst and slot of the 10-second OMEGA pattern is represented by solid vertical lines. A sequential dependence is noted by a dashed diagonal line. The execution of a function which is directly dependent upon burst and slot segmentation is shown as a solid dot, while those indirectly dependent are shown by circles.

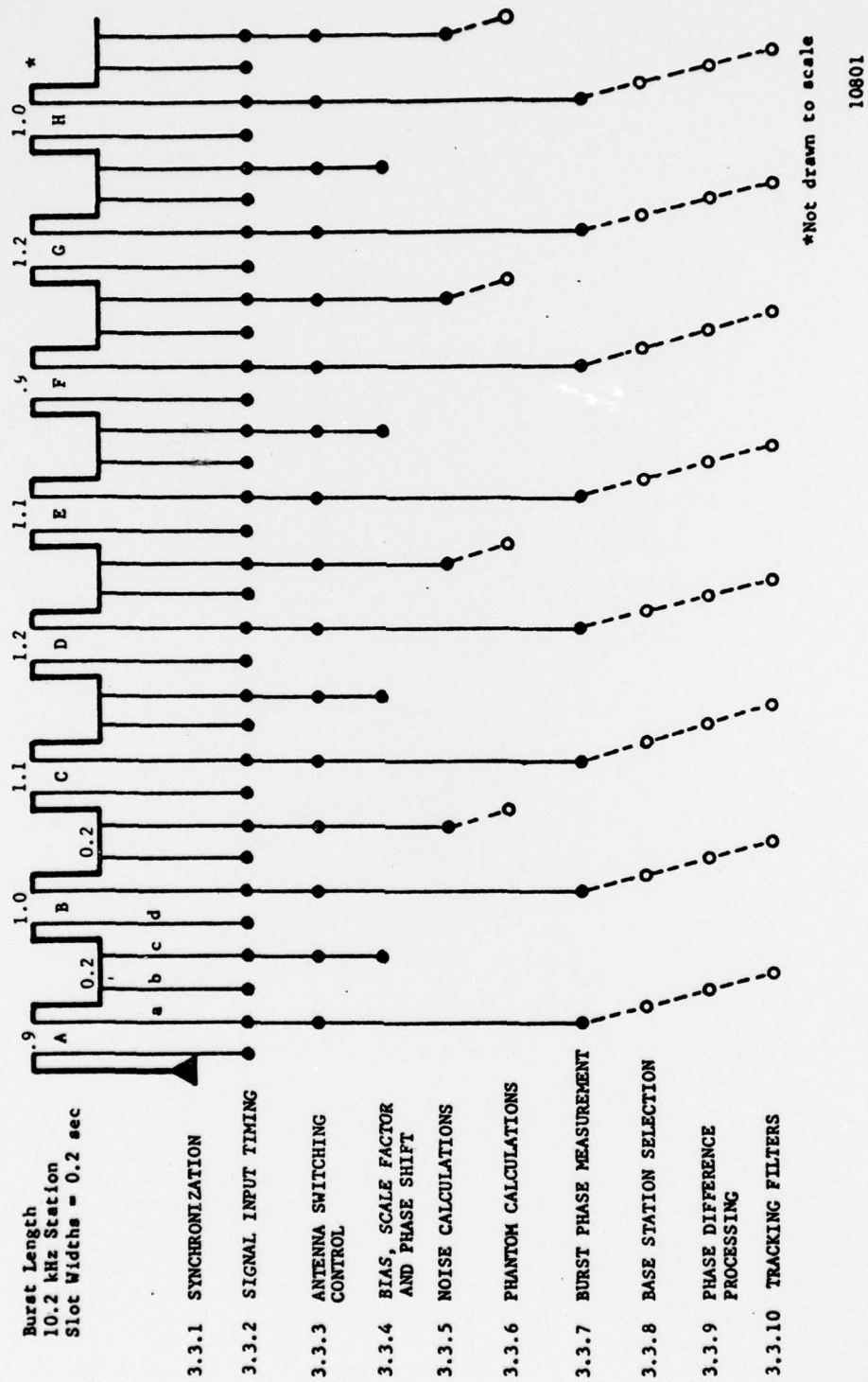


FIGURE 3 TIME & SEQUENTIAL DEPENDENCE OF FUNCTIONS TO OMEGA PATTERN

The dependence on timing of the first ten of the seventeen functional requirements is shown. The other seven have other timing requirements:

- a) The function 3.3.11 Combinational Filter is executed once in the 10-second OMEGA cycle, as is 3.3.12 Propagation Prediction. It is somewhat arbitrary where, in the cycle, these functions will be executed.
- b) The functions 3.3.13 Velocity and Heading Processing and 3.3.14 Navigation should be updated at a higher cycle rate than that of the Burst and Slot segmentation in order to minimize accumulative system error due to a lag in the velocity and position update.
- c) There are equations in 3.3.15 Control-Indicator Processing which need be iterated only if a display of the data pertaining to the equation is desired. There are other algorithms; i.e., rendezvous time and position update that will be processed at all times at a once per second frequency.
- d) The built-in test routines, 3.3.16 and 3.3.17, are also functionally unrelated to burst and slot timing and can be executed during periods of programming convenience, yet within the timing constraints of the specific test.

3.3 SUBPROGRAM DESCRIPTION AND ALLOCATION OF REQUIREMENTS

There are two categories of allocation within the Submarine OMEGA Computer Program (SOCP):

- a) Routines: a routine is a collection of functional requirements, or portions thereof, which are executed in the SOCP by one direct reference from the executive program.
- b) Subprogram: a subprogram is a collection of routines which are grouped together for the purpose of meaningful organization. A good example of the desirability for the above definitions is that of functional requirement 3.3.1 Synchronization. By programming convenience the SOCP refers to four separate routines: SYNC CALC, SYNC1, SYNC2, and SYNC3. However, the description of each individually is only meaningful in relation to the collection which is designated as the Synchronization subprogram. Table 2 shows the breakdown of functional requirements, routines and subprograms. Table 3 is the Memory Storage allocation with respect to the subprograms. An OMEGA-1070 Memory Map is included as Table 4 and will be found at the end of Section 3. An introductory flow diagram of the routines is illustrated in Figure 4. It is designated as introductory because the flow of routines cannot be understood without first understanding the executive routine and in particular the interleaved program sequencing concept.

TABLE 2 ALLOCATION OF FUNCTIONAL REQUIREMENTS TO ROUTINES AND SUBPROGRAMS

Subprogram	Routine	Functional Requirement
SYNCHRONIZATION	SYNC1	3.3.3 Antenna Switching Control Portions relating to synchronization
	SYNC2	3.3.1 Synchronization Portions thereof relating to computer start time, storage and initialization
	SYNC3	Portions of 3.3.1 relating to the collection of 0.1 second data increments
	SYNC CALC	Remainder of 3.3.1 relating to processing of the collected data
OMEGA PROCESSING	END SLOT	3.3.2 Signal Input Timing Determination of $\Delta t \Omega$ for START BURST routine 3.3.3 Antenna Switching Control Portions relating to set-up for burst measurements 3.3.4 Bias, Scale Factor and Phase Shift 3.3.5 Noise Calculations 3.3.6 Phantom Calibration
	START BURST	3.3.2 Signal Input Timing Collection of burst input data, and determination of $\Delta t \Omega$ for Combinational Filter.
	END BURST	3.3.8 Base Station Selection 3.3.2 Signal Input Timing Determination of $\Delta t \Omega$ for START SLOT routine 3.3.7 Burst Phase Measurement
	START SLOT	3.3.2 Signal Input Timing Determination of $\Delta t \Omega$ for END SLOT routine.
TRACKING FILTERS	TRACKING FILTERS	3.3.9 Phase Difference Processing 3.3.10 Tracking Filter
KALMAN	KALMAN	3.3.2 Signal Input Timing Determination of $\Delta t \Omega$ for END BURST routine 3.3.11 Combinational Filter
PROPAGATION PREDICTION	PROPAGATION PREDICTION	3.3.12 Propagation Prediction

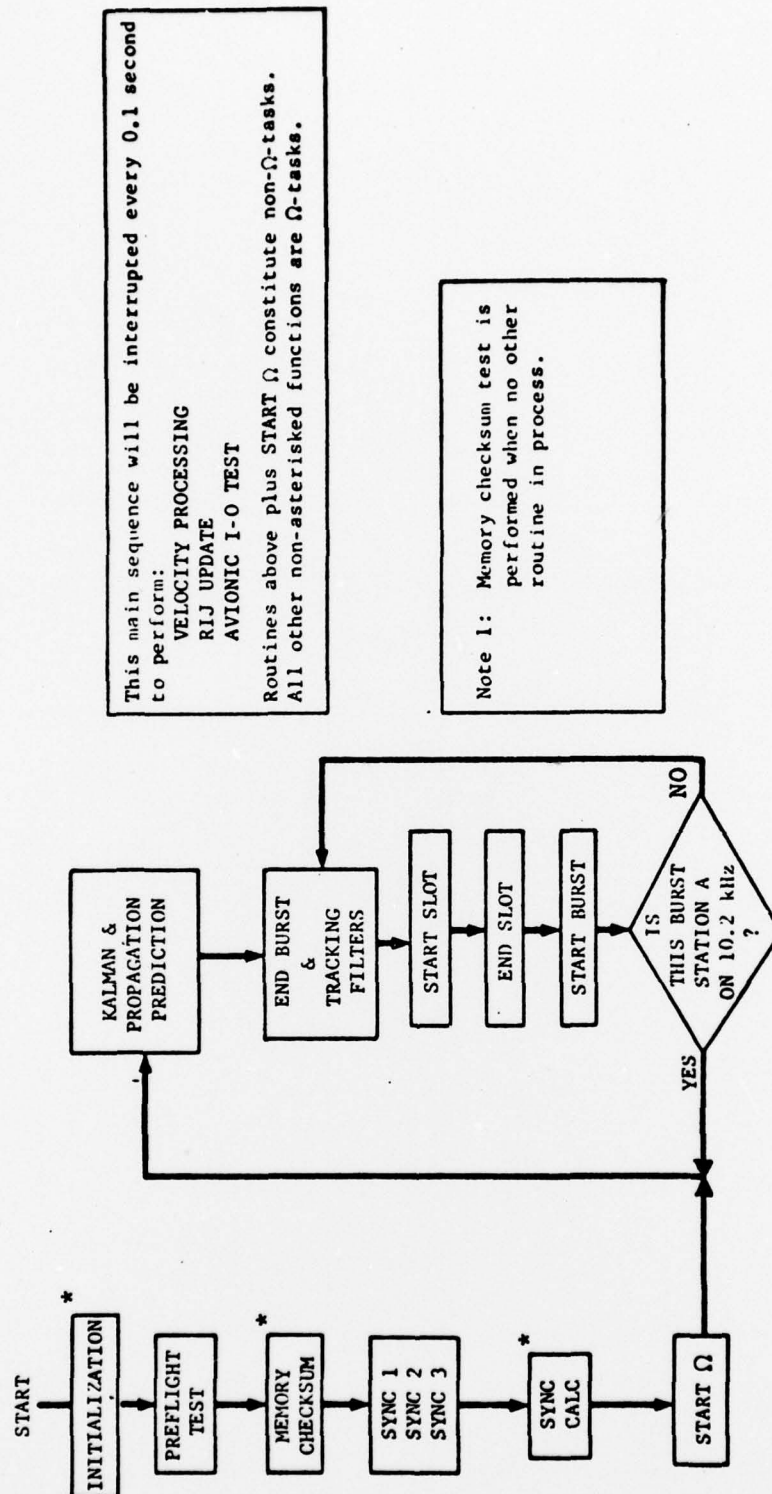
TABLE 2 ALLOCATION OF FUNCTIONAL REQUIREMENTS TO ROUTINES & SUBPROGRAMS (cont)

Subprogram	Routine	Functional Requirement
NAVIGATION	VELOCITY PROCESSING	3.3.13 Velocity and Heading processing
	RIJ UPDATE	3.3.14 Navigation
CONTROL INDICATOR	PANEL MAIN SLOW	3.3.15 Control-Indicator Procedures
	RZ UPDATE* RZ TIME UPDATE*	Equation 6 of 3.3.15.3.3 (d) Section 3.3.15.3.4(1)
TEST	GP TEST*	3.3.16 Built-in-Test
		3.3.16.4 Computer Logic Test
	AVIONIC I-O TEST	3.3.16.2 Avionic I/O Test
	RF PREAMP	3.3.16.1.2 RF Preamp
	PREFLIGHT TEST	3.3.16.1 Receiver Test
		3.3.16.6 Phase to Digital Test
		3.3.16.7 Phase Counter Test
	FREQUENCY STABILITY	3.3.16.8 Direct Memory Access Test
	MEMORY CHECKSUM	3.3.16.13 Frequency Stability Test
		3.3.16.5 Memory Checksum
COMMON	COMMON SUBROUTINES	All functional requirements using square root sine cosine arc tangent
EXECUTIVE	INITIALIZATION START OMEGA	Initialization of all requirements 3.3.2 Signal Input Timing Determination of $\Delta t \Omega$ for START SLOT routine. This is the transitional routine between 3.3.1 Synchronization and the iterated OMEGA navigation processing. This routine is discussed more fully below.
	INTERRUPTS EXEC	Discussed below Discussed below

*Indicates subroutines of the SLOW routine.

TABLE 3 OMEGA WORD-TIME STATUS

<u>Subprogram</u>	<u>Size</u>	<u>% Time</u>
Propagation Prediction	583	32.0
Tracking Filter	835	2.0
OMEGA Processing	828	0.8
Kalman	1,846	7.2
TEST	306	--
Control & Indicator	1,704	1.2
Executive	248	2.1
Navigation	213	5.9
Sync	281	--
Constants & Literals	213	--
Common Subroutines	200	--
Destination	40	--
Station Location	48	--
Conductivity Table	362	--
Push-Down Stack	189	--
TOTAL	7,896	51.2
Debug	276	0.9
TOTAL	8,172	



10803

*Controlled by Main Program; all others controlled by Five Millisecond Interrupt Routine.

FIGURE 4 INTRODUCTORY FLOW DIAGRAM OF ROUTINES

3.4 EXECUTIVE ROUTINE

Historically, the label Main Program has been given to that primary software routine that encompasses the main body of program calculation and requires a large amount of execution time. The interrupt routine is one that temporarily preempts execution of the Main Program in order to initiate a subroutine of time dependent nature. Here, however, the time dependent and sequential nature of the Submarine OMEGA software has boosted the interrupt routine to eminence and reduced Main Program to the state of marking time, or idling. Main Program will initiate the executive routine at start-up and then will continually check core until interrupted.

3.4.1 Interrupts

There are two basic types of interrupts used by the Submarine OMEGA Computer Program: those induced externally by operator intervention, and that interrupt used by the OMEGA program to maintain proper timing and sequencing of the software.

3.4.1.1 External Interrupts

- a) Power on; activated once by application of power to the OMEGA system. Normal program sequencing follows.
- b) Support Equipment; this interrupt is not used in the SOCP.
- c) System Interface Simulator; this interrupt is similar to the Power-On interrupt except that subsequent sequencing is controlled by the diagnostic routines. This interrupt is not used by operational programs.
- d) Control-Indicator; indicates panel inputs or outputs are desired by the operator. SOCP processes immediately.

3.4.1.2 Internal Interrupts

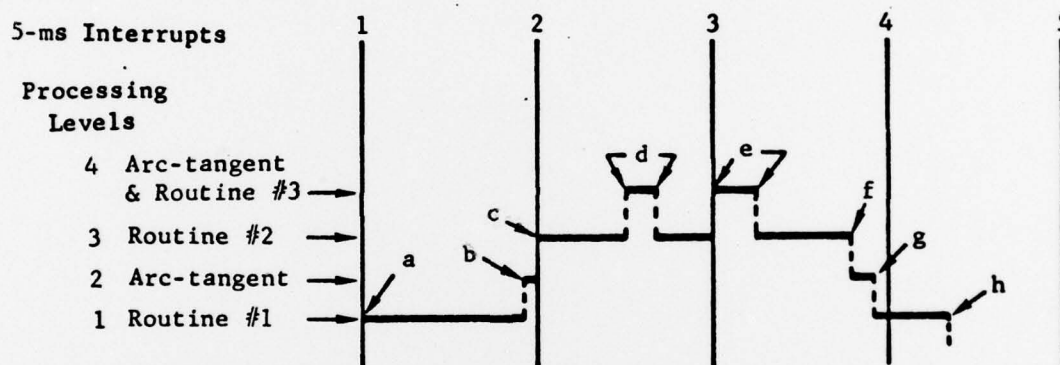
All timing and sequencing of the OMEGA Program is done by the use of the 5-millisecond interrupt originating from the Precision Frequency Generator in the receiver section of the system.

Every 5 milliseconds the computer program will halt processing of the current software task. It will then check a list of priorities to deter-

this way the software will always initiate sequencing of those routines which are dependent on the OMEGA burst and slot pattern at their scheduled initiation time.

3.4.2 Interleaved Program Sequencing

This method of sequencing routines is sometimes referred to as a "first in - last out" concept. It is a programming tool which is made convenient by the push-down stack mechanization of the advanced hardware design of the NDC-1070 computer. There are 19 processing levels to the push-down stack. A routine that is initiated on the first processing level of the roll table will be the last one finished. An example of this method of sequencing is shown in Figure 5.



- a) Interrupt routine initiates Routine #1 on processing level #1.
- b) A few microseconds before next interrupt Routine #1 uses a common subroutine, say arc-tangent.
- c) However, the arc-tangent is momentarily halted by the interrupt which initiates Routine #2.
- d) Here, Routine #2 also uses and completes the arc-tangent subroutine.
- e) Another routine, #3, is initiated by interrupt and completed.
- f) Routine #2 is completed and arc-tangent routine used by Routine #1 is continued.
- g) Arc-tangent routine completed and Routine #1 is continued
- h) Routine #1 completed.

FIGURE 5 EXAMPLE OF INTERLEAVED PROGRAM SEQUENCING

In a time dependent computer program as that of OMEGA a routine of normally short duration on the lowest (first) processing level followed by a number of long routines on higher levels could easily destroy the synchronous nature of the sequencing due to the fact that an extended time period could pass before the program returns to the lower processing level. To insure against such a situation the OMEGA Computer Program is organized such that it always puts the Combinational Filter (Kalman) followed by Propagation Prediction on the lowest processing level. By analysis of worst-case situations all other routines are of known non-conflicting durations.

Thus, at some point in time, the OMEGA program could be at some processing point within 19 routines. However, by purposeful organization it is known that there are no conflicts involving interrelated completion times of the routines. Consequently, the Five Millisecond Interrupt Routine is only concerned with initiation of tasks.

3.4.3 Main Program vs the Five Millisecond Interrupt Routine

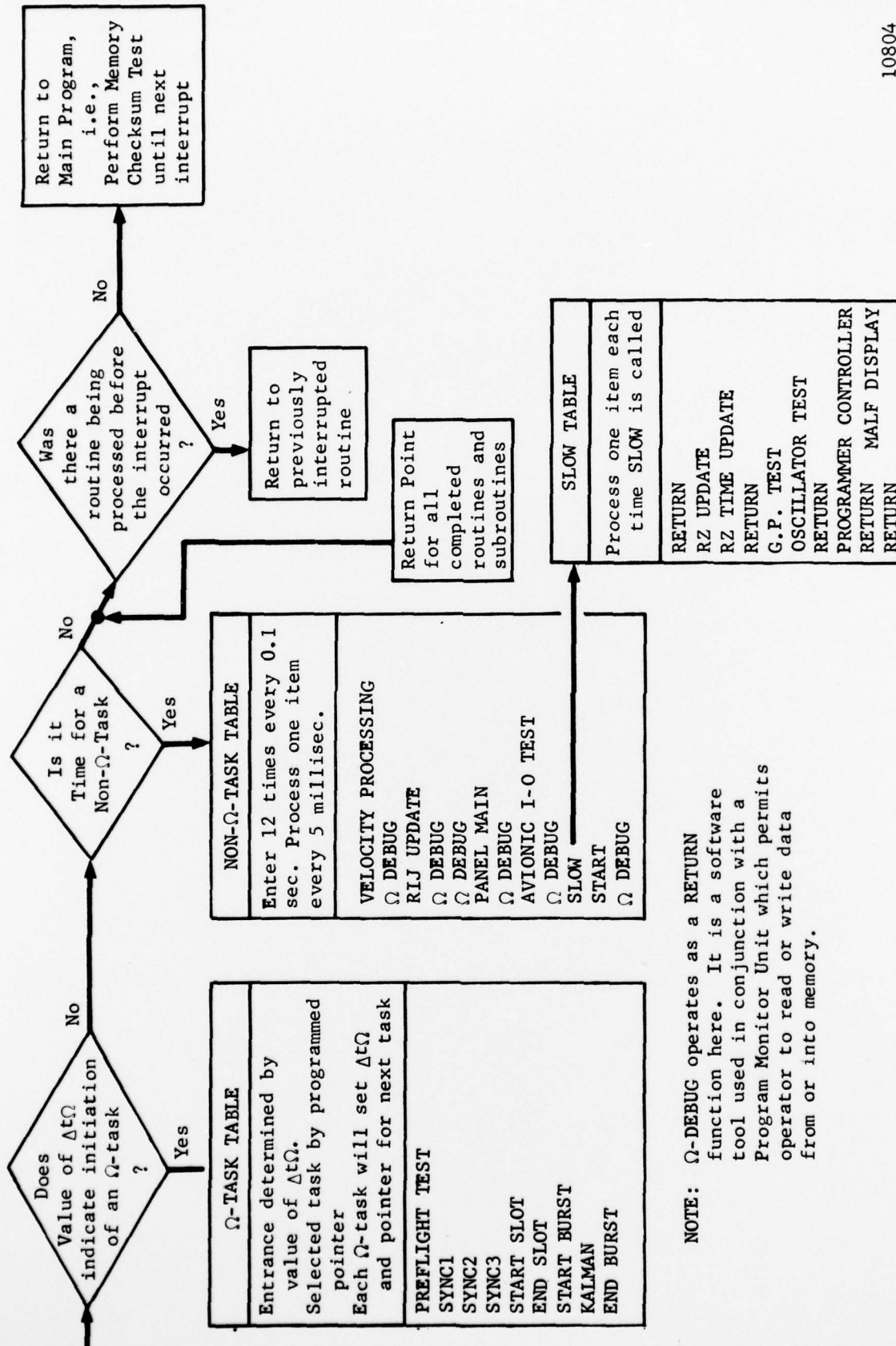
The functional flow diagram of the Five Millisecond Interrupt Routine is illustrated in Figure 6; that of the Main Program in Figure 7. To more fully understand the relationship between Main Program and the Five Millisecond Interrupt Routine it will be helpful to proceed step by step through both the start-up procedure and the main navigational iteration loop.

3.4.3.1 Start-Up

At initiation of power-on the computer interrupts are inhibited and Main Program initializes the computer program as indicated. Initialization includes setting the SYNC marker to false. The first task is Preflight Test which includes receiver, phase to digital, phase counter and DMA, GP and Avionics I/O tests. However, these tests comprise an Ω -task which is a function of the Five Millisecond Interrupt Routine. Main Program

- a) sets $\Delta t \Omega = 50$ milliseconds,
- b) selects first Ω -task to be Preflight Testing,
- c) enables the computer interrupt,
- d) performs a memory checksum (see Volume I, paragraph 3.3.16.5) and
- e) completes the enabling of the Five Millisecond Interrupt Routine by enabling interrupts in the input-output section.

The Memory Checksum test will be repeated until the interrupt routine initiates the Preflight Test as the first Ω -task.



NOTE: Ω-DEBUG operates as a RETURN function here. It is a software tool used in conjunction with a Program Monitor Unit which permits operator to read or write data from or into memory.

FIGURE 6 FLOW DIAGRAM OF THE 5.0 MILLISECOND INTERRUPT ROUTINE

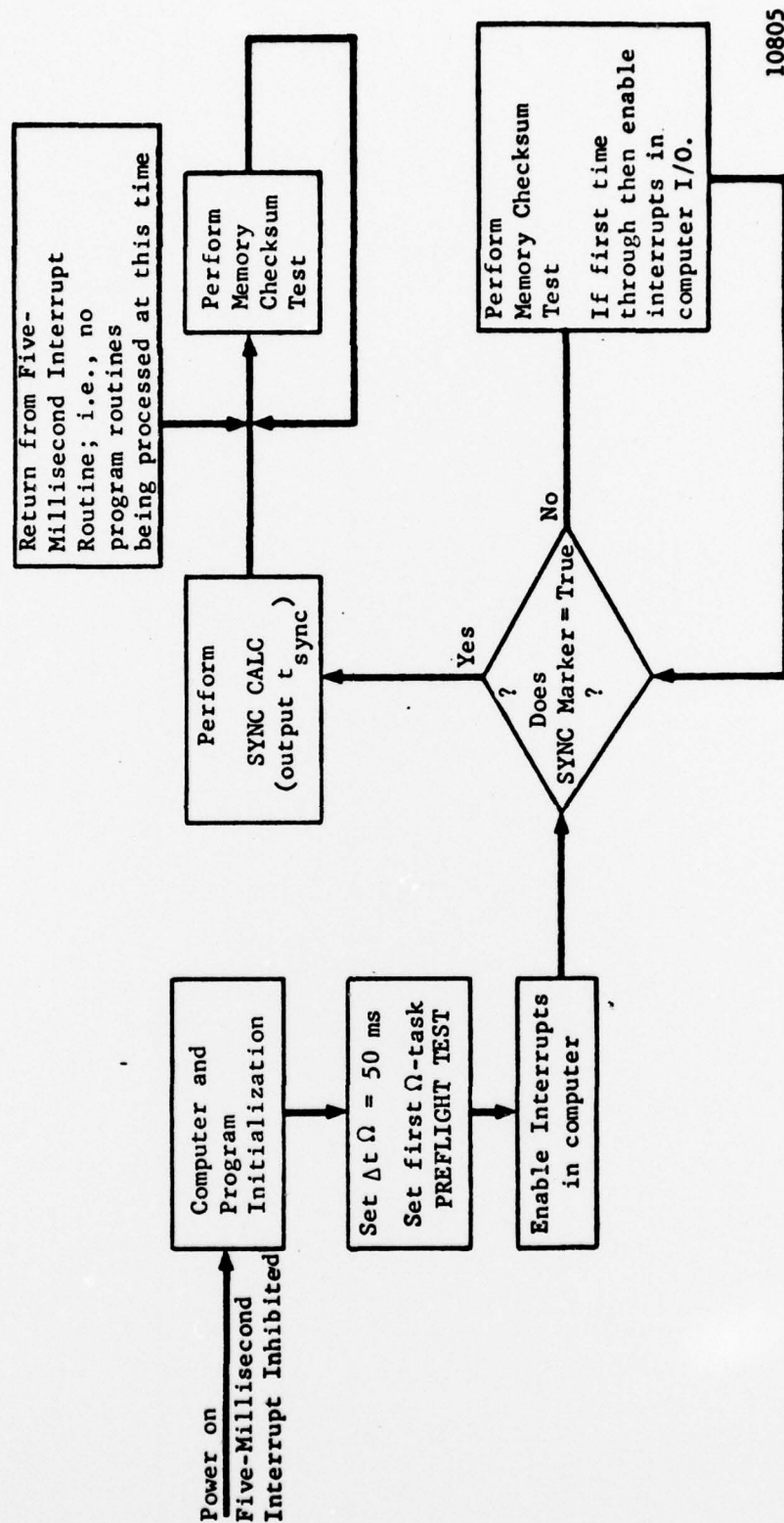


FIGURE 7 FLOW DIAGRAM OF MAIN PROGRAM

The Preflight Test routine

- a) sets $\Delta t\Omega = 0.2$ second
- b) sets the next Ω -task as Synchronization
- c) Initiates Preflight testing as described above.

As with all routines that are being processed, the Five Millisecond Interrupt Routine will cause momentary suspensions in processing the Preflight Test routine in order to perform Non- Ω -tasks. This is a continuing procedure and henceforth must be assumed when not mentioned.

Upon completion of the Preflight Tests the sequencing returns to Main Program, which performs the Memory Checksum test. This continues until the passage of $\Delta t\Omega = 0.2$ second set by the Preflight Testing routine, whereupon the next Ω -task selected by the interrupt is synchronization. Sequencing here includes:

- a) SYNC1 (sets $\Delta t\Omega = 0.2$ second)
- b) SYNC2 (sets $\Delta t\Omega = 0.1$ second)
- c) SYNC3 (data collection) set SYNC marker true, and repeat SYNC3 until marker from Main Program indicates completion of synchronization

At the setting of SYNC marker = true, Main Program performs its only task which is SYNC CALC. Upon completion:

- a) the time between the computer start time and the nearest rise time of station A burst, t_{sync} is recorded;
- b) a marker indicating completion is sent to the SYNC3 routine to halt data collection, and
- c) the START Ω marker is set true. This marker is used by the START routine which is in the Non- Ω -task table.

Henceforward the Main Program is to perform the Memory Checksum test when there are no other routines in process.

The Five Millisecond Interrupt Routine continues iterating the non- Ω -tasks, eventually processing START Ω . The first step in this routine is to test the START Ω marker, which will be true only after a synchronization processing. When the marker is false the sequencing returns as indicated in the flow diagram, to the routine previously interrupted.

When the marker is true the START Ω routine is initiated as indicated in Figure 8.

The purpose of this routine is to bridge the gap between the end of the synchronization process and the start of OMEGA navigation, and to synchronize the initiation of the non- Ω -task table with that of the Ω -task table. The objectives can be stated as follows:

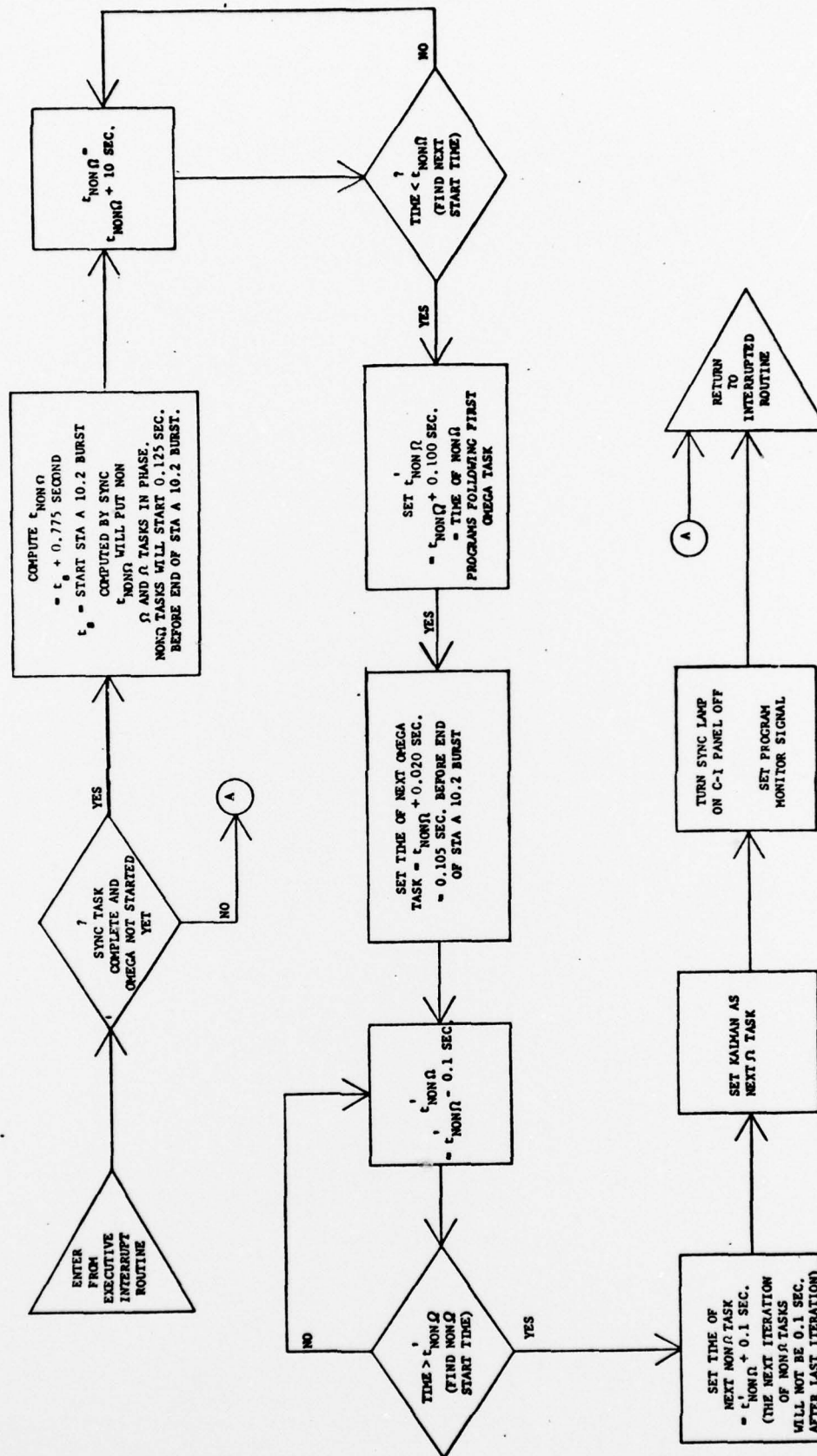


FIGURE 8 OMEGA PROGRAM - EXECUTIVE III - START OMEGA
(A NON-OMEGA TASK)

- a) Set Kalman as the next Ω -task and calculate $\Delta t \Omega$.
This insures that Kalman will be initiated on the lowest processing level.

- b) Set $t_{\text{NON-}\Omega} = \Delta t \Omega - 0.02$

This insures that R_{ij} update (position) is accomplished 5 milliseconds before Kalman processing. The purpose here is to make available to Kalman the most recent position data.

- c) Set START Ω marker = false.

This insures that the START Ω routine will not be processed during normal program iterations.

3.4.3.2 OMEGA Navigation Iteration

With these last few additional notes the sequencing of the Five Milli-second Routine is straightforward:

- a) KALMAN sets $\Delta t \Omega = 0.005$ second and the next task as END BURST. KALMAN is entered at every burst; however, an "8" counter precludes the initiation of processing until the station A burst on 10.2 kHz. PROPAGATION PREDICTION will follow KALMAN on the same processing level.
- b) END BURST sets $\Delta t \Omega = 0.15$ second and the next task as START SLOT. END BURST will automatically be followed by TRACKING FILTERS on the same processing level. TRACKING FILTERS will time update all 24 tracking filters and measurement update the 3 associated with this burst.
- c) START SLOT sets $\Delta t \Omega = 0.1$ second and the next task as END SLOT.
- d) END SLOT sets $\Delta t \Omega = 0.15$ second and the next task as START BURST.
- e) START BURST sets $\Delta t \Omega = 0.695, 0.795, 0.895$, or 0.995 second, depending upon the burst length, and sets the next task as KALMAN. This finishes one iteration of the Ω -task table.
- f) The Non- Ω -task table will be initiated once every 0.1 second. Within the table is the SLOW routine which includes five entries and five return functions. One of the ten entries will be processed for each iteration per second of Rendezvous position and time update, oscillator test, and G.P. Test.

TABLE 4 OMEGA - 1070 MEMORY MAP

Location (hexadecimal)	Contents	
- 0	Miscellaneous Switch Input	*
1 - 2	C-I Input	*
3 - D	C-I Output	*
E - E	Variable	*
F	Variable	*
10 - 16	Receiver Input	*
17 - 1F	Avionics Input	*
20 - 23	Receiver Output	*
24 - 7F	Variable	*
80 - BF	Extended Address Variable	*
C0 - D4	Extended Address Variable	
D5 - FF	Extended Address Constants	**
100 - 101	Constants	**
102 - 10D	Pin Routine Pointers	**
10E - 13F	Constants	**
140 - 141	Power On Start Address	**
142 - 143	Power Off Interrupt Address	**
144 - 145	Support Equipment Interrupt Address	**
146 - 147	5 Millisec Interrupt Address	**
148 - 149	C-I Panel Interrupt Address	**
14A - 160	Constants	**
		**
160 - 17D5	Program/Constants	**
17D6-180B	Station Locations	**
180C-181C	Constants	**
181D-1986	Conductivity Table	**
1987-1AB8	Program/Constants	**
- 1AB9	Checksum D5-19A4	**
		**
1ABA-1B25	Literals	**
1B26	Checksum 1ABA - 1B25	**
1B27-1B38	Unused (=0)	**
1B39	End of the R15 Pushdown Stack	
1B39-1BE7	R15 Pushdown Stack	*
1BE8-1BFF	F14 Pushdown Stack	*
1C00-1F7B	Variable	*
1F7C-1FC3	Variable	
1FC4-1FDD	Destination Longitudes	
1FDE-1FF7	Destination Latitude	
1FF8-1FFF	Variable	

*This area of memory is initialized to zero at turn-on.

**This area of memory is fixed and sums to zero.